UNIVERSITY OF THESSALY

MASTER OF SCIENCE THESIS

Wireless Jamming Detection in Vehicular Networks Using Machine Learning and Cross-Layer Data

Author:
Dimitrios KARAGIANNIS

Supervisors:
Dr. Antonios ARGYRIOU
Dr. Spyros LALIS
Dr. Michael VASSILAKOPOULOS

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science
Science and Technology of Electrical and Computer Engineering

in the

School of Engineering
Department of Electrical and Computer Engineering

February 12, 2018

Declaration of Authorship

I, Dimitrios KARAGIANNIS, declare that this thesis titled, "Wireless Jamming Detection in Vehicular Networks Using Machine Learning and Cross-Layer Data" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Master of Science degree at the Department of Electrical and Computer Engineering of the University of Thessaly.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF THESSALY

Abstract

School of Engineering
Department of Electrical and Computer Engineering

Master of Science
Science and Technology of Electrical and Computer Engineering

Wireless Jamming Detection in Vehicular Networks Using Machine Learning and Cross-Layer Data

by Dimitrios KARAGIANNIS

Wireless communications are vulnerable against radio frequency (RF) jamming which might be caused either intentionally or unintentionally. A particular subset of wireless networks, the vehicular ad-hoc networks (VANET), that incorporate a series of safety-critical applications, may be a potential target of RF jamming with detrimental safety effects. To ensure secure communication and defend it against this type of attacks, an accurate detection scheme must be adopted.

This work studies the detection of such attacks leveraging the use of unsupervised and supervised machine learning techniques. The machine learning algorithms, K-Nearest Neighbors (KNN), Random Forest (RF) (supervised algorithms) and K-means (unsupervised algorithm), utilize a series of features among which is the metric of the variations of relative speed (VRS) between the jammer and the receiver that is passively estimated from the combined value of the valuable and the jamming signal at the receiver. To the best of our knowledge, this metric has never been utilized before in a machine-learning detection scheme in the literature. Through clustering and classification, as well as with the utilization of the VRS feature, we are able to efficiently detect various cases of Denial of Service (DoS) jamming attacks, differentiate them from cases of interference as well as forsee a potential danger sucessfully and act accordingly.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Περίληψη

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ

Master of Science
Επιστήμη και Τεχνολογία ΗΜΜΥ

Μελέτη Ανίχνευσης Επιθέσεων με Χρήση Διαστρωματικών Δεδομένων και Τεχνικές Μηχανικής Μάθησης σε Δίκτυα Κινούμενων Κόμβων

by Δημήτριος ΚΑΡΑΓΙΑΝΝΗΣ

Οι ασύρματες επικοινωνίες είναι ευάλωτες σε επιθέσεις παρεμβολών (radio frequency (RF) jamming) που μπορεί να προκληθούν είτε σκόπιμα είτε ακούσια. Ένα συγκεκριμένο υποσύνολο ασύρματων δικτύων, τα αυτοοργανωμένα (ad-hoc) δίκτυα οχημάτων (VANET), που ενσωματώνουν μια σειρά εφαρμογών κρίσιμης σημασίας για την ασφάλεια, μπορεί να είναι ένας πιθανός στόχος παρεμβολών με επιβλαβή αποτελέσματα ασφάλειας. Προκειμένου να διασφαλιστεί η ασφαλής επικοινωνία και να προασπιστεί από αυτού του είδους τις επιθέσεις, πρέπει να υιοθετηθεί ένα ακριβές σύστημα ανίχνευσης.

Η παρούσα εργασία μελετά την ανίχνευση τέτοιων επιθέσεων αξιοποιώντας τη χρήση μη εποπτευόμενων και εποπτευόμενων τεχνικών μηχανικής μάθησης. Οι αλγόριθμοι μηχανικής μάθησης, k-Nearest Neighbors (KNN), Random Forests (RF) (τεχνικές εποπτευόμενης μηχανικής μάθησης) και K-means (τεχνική μη εποπτευόμενης μηχανικής μάθησης) χρησιμοποιούν μια σειρά από μετρικές μεταξύ των οποίων είναι η μέτρηση των μεταβολών της σχετικής ταχύτητας (VRS) ανάμεσα στον παρεμβολέα και στον δέκτη, η οποία εκτιμάται παθητικά από τη συνδυασμένη τιμή του πολύτιμου και του σήματος παρεμβολής στον δέκτη. Από όσο μας είναι γνωστό, αυτή η μετρική δεν έχει χρησιμοποιηθεί ποτέ πριν σε αντίστοιχη προσπάθεια ανίχνευσης επιθέσεων παρεμβολής με χρήση μηχανικής μάθησης στη βιβλιογραφία. Μέσω της ομαδοποίησης και της ταξινόμησης, καθώς και με τη χρήση της προτεινόμενης μετρικής VRS, είμαστε σε θέση να ανιχνεύσουμε αποτελεσματικά διάφορες περιπτώσεις επιθέσεων παρεμβολής με στόχο την άρνηση υπηρεσίας (DoS επιθέσεις), να τις διαφοροποιήσουμε από περιπτώσεις ακούσιας παρεμβολής καθώς και να προβλέψουμε επιτυχώς έναν πιθανό κίνδυνο ώστε να ενεργήσουμε αναλόγως.

Acknowledgements

Concluding this academic effort, i would like to express my gratidute towards those who helped me and supported me.
First and foremost, i would like to sincerely thank my supervisor, Assistant Professor Mr. Antonios Argyriou, for his guidance, his support and for all the time he dedicated to me during the course of this thesis. I would, also, like to thank PhD candidate Mr. Dimitrios Kosmanos for his cooperation and advices and Associate Professors Mr. Spyros Lalis and Mr. Michael Vassilakopoulos for participating in the evaluation comittee.
Last, a big thank you is owed to my family and friends, without the support, love and encouragement of which i could not have completed this journey.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
AWGN	**A**dditive **W**hite **G**aussian **N**oise
CACC	**C**ooperative **Adaptive C**ruse **C**ontrol
CBR	**C**hannel **B**usy **R**atio
CC	**C**orrelation **C**oefficient
CRC	**C**yclic **R**edundancy **C**ontrol
DoS	**D**enial **o**f **S**ervice
DSRC	**D**edicated **S**hort **R**ange **Communication**
EM	**E**lectro **M**agnetic
EP	**E**rror **P**robability
FN	**F**alse **N**egative
FNR	**F**alse **N**egative **R**ate
FP	**F**alse **P**ositive
FPR	**F**alse **P**ositive **R**ate
IDE	**I**ntegrated **D**evelopment **E**nvironment
IOT	**I**nternet **O**f **T**hings
IOV	**I**nternet **O**f **V**ehicles
IT	**I**nactive **T**ime
KNN	**K N**earest **Neighbors**
KNN-VRS	**KNN**-based classification model with the **VRS** feature
LOS	**L**ine **O**f **S**ight
MANET	**M**obile **A**d-hoc **N**etwork
ML	**M**achine **L**earning
MFM	**M**odified **F**ading **M**emory
PDR	**P**acket **D**elivery **R**atio
PDSR	**P**acket **D**elivery (and) **S**end **R**atio
PLR	**P**acket **L**oss **R**atio
PSR	**P**acket **S**ending **R**atio
RF	**R**adio **F**requency
RF-VRS	**RF**-based classification model with the **VRS** feature
RSSI	**R**eceived **S**ignal **S**trength (and) **I**nterference
RSU	**R**oad **S**ide **U**nit
SINR	**S**ignal (to) **I**nterference (and) **N**oise **R**atio
SVM	**S**upport **V**ector **M**achine
TN	**T**rue **N**egative
TNR	**T**rue **N**egative **R**ate
TP	**T**rue **P**ositive
TPR	**T**rue **P**ositive **R**ate
VANET	**V**ehicular **A**d-hoc **NET**work
VRS	**V**ariations (of) **R**elative **S**peed

V2R	**V**ehicle **2** (to) **R**oadside (unit)
V2V	**V**ehicle **2** (to) **V**ehicle
WAVE	**W**ireless **A**ccess (in) **V**ehicular **E**nvironment

Dedicated to my family and friends…

Chapter 1

Introduction

1.1 Vehicular Communications

Autonomous vehicles capable of navigating unpredictable real-world environments with little human feedback are a reality today [1]. Autonomous vehicle control imposes very strict requirements on the security of the wireless communication channels [11], [30] used by the fleet of vehicles, in order for them to exchange information and remain connected [5]. The Intelligent Vehicle Grid technology is introduced in [9], in which the car becomes a formidable sensor platform, absorbing information from the environment or from other cars (and from the driver) and feeding it to other vehicles and infrastructure so as to assist in safe navigation, pollution control and traffic management. The Vehicle Grid essentially becomes an Internet of Things (IOT), which we call Internet of Vehicles (IOV), capable of making its own decisions in driving customers to their destinations [12].

Wireless communications, however, are vulnerable against a wide range of attacks [20]. An attack that is particularly hard to detect in every wireless network is the RF jamming attack [7]. In a VANET, attack detection is even more difficult due to the constant and rapid changes in topology and the high mobility of the nodes as well as due to the presence of a variety of different jammers [23] affecting either the communication between vehicles (V2V communication) or the communication between the vehicles and the roadside units, namely RSUs (V2R communication).

Over the last few years, there have been several experimental approaches for jamming detection [3], [7], [16], [17], [18], [19], [20], [32], some of which suggest the use of machine learning [6], [19]. All the above works that propose machine-learning based schemes, do not utilize the relative speed, which is a metric of the application layer. In a few cases this metric is used in the training procedure with it's value being measured by sensors. The estimation of the relative speed metric from the wireless medium is not used from previous works mainly because of the fact that vehicular wireless channels exhibit specific characteristics (i.e. rapidly changing vehicular channels) that make them quite different from the very well defined mobile telephony channels. However with our work, we prove that this metric can be utilized in a realistic scenario with a minimum number of assumptions.

1.2 Thesis Motivation and Contribution

The successful and on-time apprehension and avoidance of a real jamming attack could prove to be crucial in an urban environment for the preservation of the safety. Additionally, the differentiation between intentional jamming and interference - that is unintentional jamming - is also very important as a different defencive behavior could be adopted in each case. In the case of interference, an Interference Cancellation (IC) model [2], a spectral evasion (channel surfing) or a spatial evasion (spatial retreats) scheme combined with adjusting resources, such as power levels and communication coding, could be adopted to preserve communication [24]. On the other hand, in the case of a jamming attack, a different approach, such as the Hideaway Strategy [3] must be adopted as the previously described solutions would not be effective against a real and persistent jammer. For the validation of the proposed approach, one interference-only scenario and two jamming attack scenarios have been created and tested.

This work focuses on the detection of possible RF jamming attacks - aiming at disrupting the communication of the nodes in a VANET - in an urban area and their differentiation from unintentional jamming (interference), using supervised and unsupervised machine learning techniques, that, as it is already stated, has not been closely examined by previous works. Apart from the utilization of machine leraning, we use a series of cross-layer metrics in addition to our new, variations of relative speed (VRS), metric, so as to make the detection process more robust agaist different types of jamming, intentional or unintenional.

The contribution of this thesis is three-fold:

- A novel detection scheme is introduced that leverages a new metric, namely *the variations of relative speed (VRS)* that, to the best of our knowledge, has never been used in literature before in supervised or unsupervised machine learning-based jamming attack detection approaches.
- A completely passive scheme is utilized using the combined received signal at the receiver, without hardware or computational cost. The scheme first estimates the combined channel of the transmitter - receiver ($T_x - R_x$) and second the channel of the jammer - receiver ($J_x - R_x$). It then estimates the relative speed between the jammer and the receiver using the RF Doppler shift.
- Based on a series of cross-layer data (among which is the VRS metric that is calculated using the estimated relative speed) detection of a potential threat as well as differentiation between a case of jamming and a case of interference is achieved.

1.3 Structure

The rest of this thesis is structured as follows: Chapter 2 provides an overview of the related work in the domain of attack (not only jamming) detection

as well as some basic concepts, Chapter 3 describes the adopted topology and the channel model, Chapter 4 presents the proposed machine-learning based jamming detection system for both supervised and unsupervised approaches, Chapter 5 describes the simulation setup, Chapter 6 presents the experimental results and comparisons and Chapter 7 summarizes the significance of our approach and concludes this work.

Chapter 2

Related Work and Basic Concepts

2.1 Related Work

The most important machine-learning based approaches for RF jamming attack detection in vehicular ad-hoc networks have been reported in [19] and [6].

Grover et al. [6] propose a machine learning approach to classify multiple misbehaviors in VANET using concrete and behavioral features for each node that sends safety packets. However, features related to the verification of geographical position, such as speed, are only required to classify position and identity spoofing (Sybil) attacks.

Puñal et al. [19] use several channel- Noise and Channel Busy Ratio (CBR), performance - Packet Delivery Ratio (PDR) and Maximum Inactive Time (Max IT)- and signal- Received Signal Strength (RSS)- metrics to perform an attack detection utilizing machine learning techniques and examining the cases of reactive and constant jammers.

Azogu et al. [3] have implemented a new mechanism, called Hideaway Strategy which uses the Packet Sending Ratio (PSR) and according to which all nodes should remain silent while the network is under a jamming attack.

Bißmeyer et al. [4] propose a detection scheme that is based on the verification of vehicle movement data and on the notion that a certain space will be occupied by only one vehicle at a certain time.

Hamieh et al. [7] focus on the detection of the reactive jammer. The proposed method, is based on the correlation coefficient (CC) and the error probability (EP). Each node compares the calculated value of CC with the EP and the network is considered under a jamming attack if CC>EP.

Malebary et al. [15] propose a two-phase jamming detection method that utilizes metrics such as the RSS, the Packet Delivery/Send Ratio (PDSR) and the Packet Loss Ratio (PLR) as well consistency checks to distinguish a jamming from a no-jamming situation.

Mokdad et al. [16], [17] propose a scheme for detecting a jamming attack in vehicular ad-hoc networks that depends on the variations of the PDR. The approach is based on the premise that only packets that derive from the sender are allowed through the Cyclic Redundancy Check (CRC) and that the PDR is equal to the ratio of these packets and the total number of packets received.

Puñal et al. [18] create a set of jammers and implement a variety of jamming scenarios, both indoor and outdoor, under different jamming behaviors

(constant, reactive and pilot jamming) in order to address the impact of a RF jammer in VANET communications.

Quyoom et al. [20] and RoselinMary et al. [21] present an approach that is based on the detection of malicious and irrelevant packets using the number of broadcast packets per second (frequency) and the velocity of the vehicle that the packets are sent from.

Shafiq et al. [22] propose an attack detection approach based on the number of packets that are received from each vehicle, thus indicating an attack if this number is greater than the threshold value.

Xu et al. [32] state the inability of the PDR to differentiate jamming from interference cases. For that reason, two detection schemes are proposed. The first one utilizes signal strength measurements as a consistency check to determine if the PDR value is due to jamming or interference. The second uses location information as the consistency check. Several jamming attack models are presented and evaluated.

Amoozadeh et al. [1] study the case of Cooperative Adaptive Cruise Control through the illustration and simulation of a series of different layer attacks as well as their countermeasures, that can affect the VANET communication of the connected vehicles.

Sharanya et al. [25] propose the use of the Support Vector Machine (SVM) algorithm with Modified Fading Memory (MFM) so as to classify legitimate and malicious nodes. The purpose of the MFM is to reduce the computational overhead for the machine learning algorithm by only considering as eligible nodes those in the range of the VANET communication.

In all the prior works in which machine-learning based schemes are proposed, the metric of the relative speed and its variations is not utilized. In a few cases this metric is utilized in training with it's value being measured by sensors. In our work, the estimation of the relative speed is achieved with the use of the RF communication between jammer and receiver, the variations of which are used as an extra feature in the classification and clustering process.

2.2 VANET

VANET can be described as an alteration of the Mobile ad-hoc network (MA -NET). Their main difference is that in VANET the communicating nodes are vehicles and roadside units (RSU), thus they are characterized by constant and rapid changes in topology as well as high mobility. The main motivation behind the implementation and adoption of vehicular communication systems, is the augmentation of the road safety, the avoidance of collisions, the reduction of delays and of the environmental pollution and the overall improvement of the driving conditions.

To achieve that, uninterrupted communication and information exchange between vehicles (V2V communication) or between vehicles and roadside units (V2R communication) must be ensured. The communication, both between vehicles and between vehicles and infrastructure, is achieved making use of the IEEE 802.11p, which is an amendment to the 802.11 standard that

operates in the band of 5.9 GHz. It supports travelling speeds of up to 200 km/h, transmission range up to 1000 meters and data transfer rate up to 27 Mb/s (the default data transfer rate is 6 Mb/s).

2.3 RF Jamming

Radio Frequency (RF) jamming is the act of intentionally transmitting a signal that does not comply with legitimate physical and MAC layer protocols. The primary goal of the jammer is to disrupt or alter the communication between the transmitter and the receiver. While interference is a common form of RF jamming that can affect wireless communications - especially in an urban environment such as the one we examine - it can be distinguished from actual jamming due to the fact that is is done unintentionally because of a possible device malfunction.

2.4 Machine Learning

Machine Learning (ML) is a field of computer science, specifically a field of Artificial Intelligence (AI), that enables computers to gain knowledge and act without being explicitly programmed. This makes possible the creation of systems that interact with the environment they operate in, learn from it and improve in the way they perform a certain process.

Depending on whether or not there is a "learning signal" or "feedback" available to the system, the machine learning can be divided into two general catgories:

- *Supervised Learning*: the algorithm is trained using a number of known inputs with their corresponding outputs (training) and its purpose is to generalize the process for mapping inputs with unknown outputs (testing).
- *Unsupervised Learning*: the algorithm is responsible for finding structure in the input data, on its own, without any previous training.

Chapter 3

System Model and Relative Speed Estimation

3.1 System Model

3.1.1 Rician Fading Model

In our work, we explore the Rician fading model, that is a channel model which includes path loss and also Rayleigh fading [29]. When a signal is transmitted, whether it is a useful signal or a jamming one, this channel adds fading in addition to thermal noise. The baseband signal at the receiver is:

$$y = (h_1 + \frac{1}{d_s^2}) \times x_s \times P_s + (h_2 + \frac{1}{d_j^2}) \times x_j \times P_j + w \tag{3.1}$$

In the above h_1, h_2 are complex Gaussian random variables capturing the Rayleigh fading between transmitter - receiver ($T_x - R_x$) and jammer - receiver ($J_x - R_x$) respectively. The x_s, x_j are the symbols that are transmitted (from the transmitter and the jammer), which in our case are equal to -1 or $+1$ because we assume BPSK modulation. This modulation scheme is the most robust in a high interference environment as it uses two phases which are separated by 180°. P_s and P_j are the transmission power per symbol of the useful and of the jamming signal respectively and w is the channel noise. The terms d_s, d_j correspond to the distance between the transmitter and the receiver and between the jammer and the receiver.

3.1.2 Topology

The topology we adopt in our work (Fig. 3.1) involves a moving vehicle, namely R_x, that serves as the target of the jammer, another vehicle or a RSU (namely T_x) that is used as the transmitter of the useful signal and the jamming vehicle that tries to intervene in the communication between R_x and T_x. In our work we will assume V2V communication, thus the transmitter will also be a vehicle. The R_x travels at a speed (u_{Rx}), that is bound to the limitations of an urban environment, while communicating with the T_x. Upon spotting its target, the jammer begins following it and starts jamming either continuously or smartly (in order to stay undetected for as long as possible). In smart jamming, the jammer only transmits when sensing energy above a

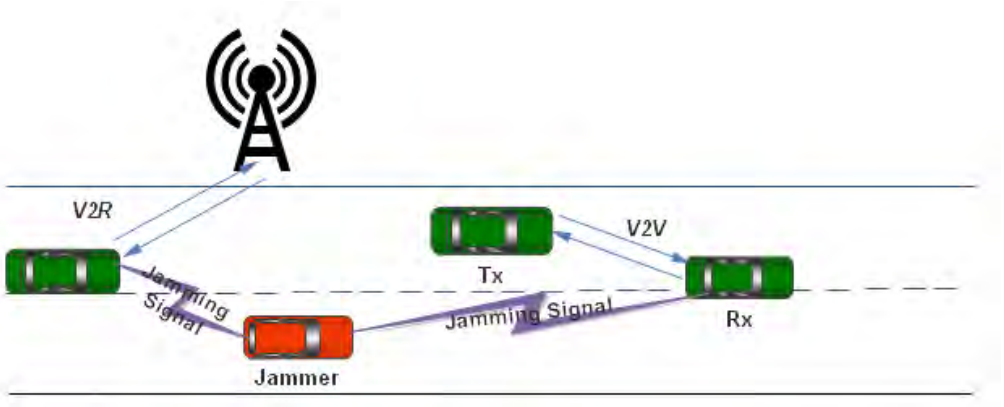

FIGURE 3.1: Topology

certain threshold at the physical layer, while in constant jamming the signal from the jammer is continuously transmitted.

3.2 Relative Speed Estimation

In the model of Fig. 3.1 we want, first, to estimate the speed of a moving jamming source in the area. We assume that the transmitter sends known pilot packets at the receiver to estimate the wireless channel (h_1) between T_x and R_x. The RF communication between $T_x - R_x$ and $J_x - R_x$ is exploited in order to, first, estimate the combined channel between the transmitter and the receiver and the channel between the jammer and the receiver (using a MMSE estimator). Next the relative speed between the jammer and the receiver is estimated using the RF Doppler shift. Subsequently, the variations of the estimated relative speed are used as a new feature for our proposed supervised machine learning approach for jamming attack detection.

Chapter 4

Proposed Detection System

4.1 Initial VRS Algorithm

To make our detection method robust, apart from some of the metrics used as features in related works (e.g. [19], [6]), we introduce and use an application layer metric, namely the VRS metric, that exploits the estimated relative speed from Section 3.2 and its variations. Our goal is to evaluate whether this new metric improves the detection results under various scenarios without adding extra complexity to our model.

Our method uses this new metric, as an extra feature, along with other, cross-layer, metrics such as the Received Signal Strength and Interference (RSSI), the Signal to Interference and Noise Ratio (SINR) and the Packet Delivery Ratio (PDR), which are jointly processed by two classification algorithms, namely the k-Nearest Neighbors (KNN) [27] and the Random Forests algorithm [14] respectively and one clustering algorithm, the K-means algorithm [8].

To create the VRS metric that will be used in the supervised and in the unsupervised learning process we, initially, made the following three fundamental assumptions:

- When the relative speed is equal to zero and remains unchanged, it indicates the existence of a jammer that follows the victim-vehicle.
- When the relative speed is not equal to zero and remains unchanged, it indicates the absence of a jammer as the relative speed is equal to the speed of the vehicle.
- When the relative speed is not equal to zero for a period of time and then becomes zero while remaining unchanged, it indicates the existence of a jammer that begins following the target after reaching it.

Based on these basic assumptions, we developed an algorithm, which was first introduced in the unsupervised learning-based RF attack detection approach, that depending on the variations of the relative speed, was able to categorize the observations in one of the three scenarios introduced in section 5.1.

Algorithm 1 Relative Speed Variations Algorithm

```
1: M = number of observations
2: VRS = matrix(nrow = M, ncol = 1)
3: while (i < M) do
4:     if Δu[i] ≠ 0 then
5:         if Δu[i] ≠ Δu[i-1] then
6:             VRS ← A
7:         else if Δu[i] == Δu[i-1] then
8:             VRS ← NA
9:         end if
10:    else if Δu[i] == 0 then
11:        VRS ← A
12:    end if
13: end while
```

Algorithm 1 creates a new metric that is based on the variations of the relative speed and is represented by the *VRS array*. Specifically, it consists of two main *if* branches so that the potential presence of a jammer may be identified by observing whether or not the relative speed is equal to zero, while considering the fact that the speed of the participating vehicles remains unchanged as the time progresses.

Having relative speed values that are constant and not equal to zero is a clear indication of the absence of a jammer, following our initial assumption of unchanged speed during the simulation. On the other hand, if the relative speed is zero then a jammer is present and follows its target. Based on that observation, the algorithm iterates, initially, through all the values of the relative speed - not equal to zero - from the Δ_u array.

Each entry of the array is compared with the previous one. If a change is observed, a value equal to A is assigned to the VRS array, thus indicating a possible attack. If the relative speed remains unchanged then a value equal to NA is assigned. The NA and A values are two extreme and distinct values able to differentiate attack from no attack cases. Moving on to the second *if* branch, the values of the Δ_u array that are equal to zero indicate a jamming attack, thus a value of A is, always, inserted into the *VRS* array.

4.2 Updated VRS Algorithm

In the previous section, the first and most basic form of the algorithm - used so as to estimate the variations of the relative speed - was introduced. The common characteristic of these assumptions is that the speed of the participating vehicles remains unchanged and is always greater than zero.

However, in a real-life scenario, such as the one that we study, the speed - and consequently the relative speed - may not remain constant during the observation. If we want to fully simulate an urban environment, we have to consider the fact that the vehicles, at some point, may need to alter their travelling speed or even completely stop. To handle these real-life situations,

while still using the previously presented assupmtions as a basis, we implemented the **Updated Variations of Relative Speed Algorithm** (Algorithm 2) with the assumption that the jammer adopts the same driving behavior as its target (i.e when the target decelerates the jammer also decelerates).

Algorithm 2 Updated VRS Algorithm

```
1:  M = number of observations
2:  vrs = matrix(nrow = M, ncol = 1)
3:  k = 1
4:  if Δu[k] == Δu[k+1] then
5:      vrs ← NA
6:      trigger = 0
7:  else if Δu[k] ≠ Δu[k+1] then
8:      vrs ← A
9:      trigger = 1
10: end if
11:
12: k++
13: while (k < M) do
14:     if Δu[k] ≠ 0 then
15:         if Δu[k] ≠ Δu[k-1] then
16:             if Δu[k] == u[k] then
17:                 vrs ← NA
18:                 trigger = 0
19:             else if Δu[k] ≠ u[k] then
20:                 vrs ← A
21:                 trigger = 1
22:             end if
23:         else if Δu[k]==Δu[k-1] then
24:             if Δu[k] ≠ u[k] then
25:                 vrs ← A
26:                 trigger = 1
27:             else if Δu[k] == u[k] then
28:                 if hasNext == T then
29:                     if (Δu[k-1]==u[k-1]&& Δu[k+1]==u[k+1]) then
30:                         vrs ← NA
31:                         trigger = 0
```

Algorithm 2 Updated VRS Algorithm (continued)

```
                else
                    vrs ← A
                    trigger = 1
                end if
            else if hasNext == F then
                if trigger == 0 then
                    vrs ← NA
                    trigger = 0
                else
                    vrs ← A
                    trigger = 1
                end if
            end if
        end if
    else if Δu[k] == 0 then
        if u[k] ≠ 0 then
            vrs ← A
            trigger = 1
        else if u[k] == 0 then
            if Δu[k-1] == u[k-1] then
                if trigger == 0 then
                    vrs ← NA
                    trigger = 0
                else
                    vrs ← A
                    trigger = 1
                end if
            else if Δu[k-1] ≠ u[k-1] then
                vrs ← A
                trigger = 1
            end if
        end if
    end if
end while
return vrs
```

The *VRS Algorithm* detects changes in the relative speed of the provided observations. To ensure that, the relative speed and the speed from the previous as well as the subsequent observations are used along with a series of control flow statements. The algorithm is divided into two main parts, with the first examining the case in which the relative speed value is not equal to zero and the second examining the opposite case, each one with its own logical checks to determine the existence of a threat or not.

Apart from the estimated relative speed, in order to handle cases of speed alterations, the speed of the receiver has to be examined as well. If Δu is not

equal to zero, then either there is no jammer present (and only interference may potentially affect the wireless communication) or there is a jammer that has not yet reached the receiver. To identify in which case we are into, we have to examine whether or not there has been a variation in the relative speed compared to a previous time instance.

Observing a variation in the relative speed, however, is not, by iteself, a clear indicator of the presence or absence of a jammer. For that reason the speed of the receiver u is, also, exploited. The equality between the relative speed (Δu) and the speed (u), while Δu changes, indicates the absence of a jammer, as the speed of the jammer u_{Jx} is equal to zero and the speed of the receiver u_{R_x} is in fact the relative speed. On the contrary, a difference between Δu and u indicates the presence of a jammer that follows the receiver.

On the other hand, if no alteration of the relative speed is observed while the relative speed value is not equal to the speed value, a possible presence of a jammer is registered. This could occur in a situation where the target vehicle would reduce its speed due to an obstacle. Following our assumption, the jammer would, also, decrease its travelling speed, thus keeping the relative speed unchanged but also different from the travelling speed of the receiver. Counter to the previous, if no alteration in the relative speed value is observed (for the previous and the next measurement) while having $\Delta u == u$, we conclude to that a jammer is not following the receiver.

Having examined the case where the observed relative speed value is not equal to zero, we proceed to the opposite case. With $\Delta u == 0$, the initial form of the algorithm (Algorithm 1), would have indicated the existance of a jammer that has reached its target and that follows it closely with the same speed. Examining a real-life environment, however, is more complicated. If the travelling speed u of the receiver is not equal to zero, while $\Delta u == 0$, a jammer has reached the receiver and follows it while disrupting the communications. On the hand, if the travelling speed is zero (while $\Delta u == 0$) there might be a jammer present that has stopped moving (following the behavior of the target). In that case, we have to examine the previous observation for equality between relative speed and travelling speed as well as the *trigger* value to determine the outcome.

The Δu and u variables represent an array of estimated relative speed values and travelling speed values respectively, M is the number of the available observations upon which the algorithm operates, *vrs* is an array used to store the estimation result (A for attack or NA for not attack) of the current observation and the *trigger* is a binary variable which indicates the presence of a jammer (value is equal to 1) or its absence (value is equal to 0).

4.3 Comparison of the VRS Algorithm Versions

The difference between the two versions of the proposed algorithm, for creating our novel metric, is evident and concerns the handling or not of the speed variations that might occur in an real-life, urban environment like the one we examine. Therefore, our detection scheme depends on the updated version of the VRS Algorithm (Algorithm 2).

Before proceeding to a following chapter, a comparison could be made using the two versions of the VRS Algorithm in unsupervised learning (clustering), utilizing the K-means algorithm and using a number of two clusters. Using a number of $k = 2$ clusters, will indicate the existance or not of a jammer (*in other words, it will differenctiate the Interference Scenario from the other two jamming attack scenarios described in section 5.1*) in a real-life situation where speed variations are possible.

4.3.1 Updated VRS Algorithm Clustering Results

From Table 4.3.1 the crusial role of the updated VRS Algorithm in handling speed variations during the simulations is evident. More specifically, from each of the three scenarios we obtain 1000 measurements and we try to categorize them under two major classes, the **Attack** and the **Interference** class respectively. [1]

Scenario	Interference Scenario	Smart Attack Scenario	Constant Attack Scenario
Attack	1	1000	1000
Interference	999	0	0

TABLE 4.1: Clustering results for the updated VRS Algorithm

By the manner in which the scenarios are created in 5.1, we, ideally, expect the observations of the two jamming scenarios, that is 2000 observations in total, to be categorized in the "Attack" class, while the remaining 1000 observations from the interference scenario to be categorized in the "Interference" class. The updated VRS Algorithm produces a clustering result close to the ideal one, with only one interference observation being wrongfully categorized as an attack observation (Fig. 4.1).

4.3.2 Initial VRS Algorithm Clustering Results

Without the utilization of the updated VRS Algorithm, however, the clustering results obtained when attempting to examine the same observed values from the three scenarios, are highly confused. Due to the fact that the initial VRS Algorithm was designed to operate under optimal circumstances, many of the observations are wrongly classified, thus an unreliable scheme, when dealing with real-life circumstances, is created (Fig. 4.2).

[1] Although the term "categorize" is not utterly correct when considering a clustering problem, we will use it so as to present the group in which the observations are predicted to belong to.

Scenario	Interference Scenario	Smart Attack Scenario	Constant Attack Scenario
Attack	690	679	894
Interference	310	321	106

TABLE 4.2: Clustering results for the initial VRS Algorithm

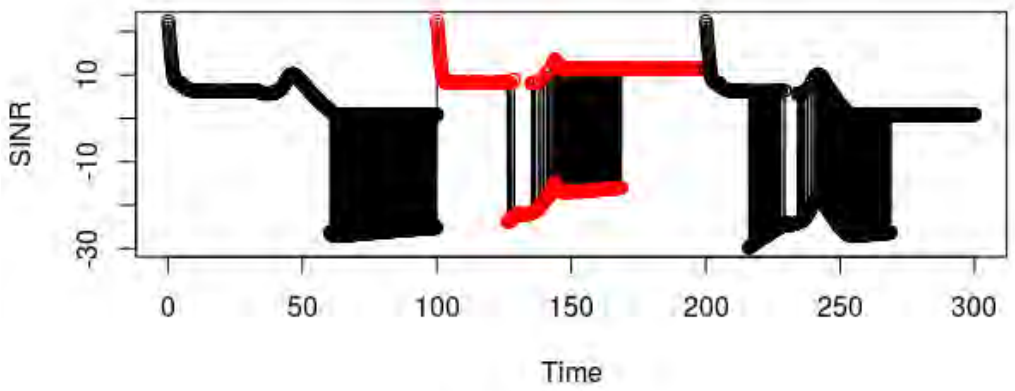

FIGURE 4.1: Clustering results for the updated VRS Algorithm

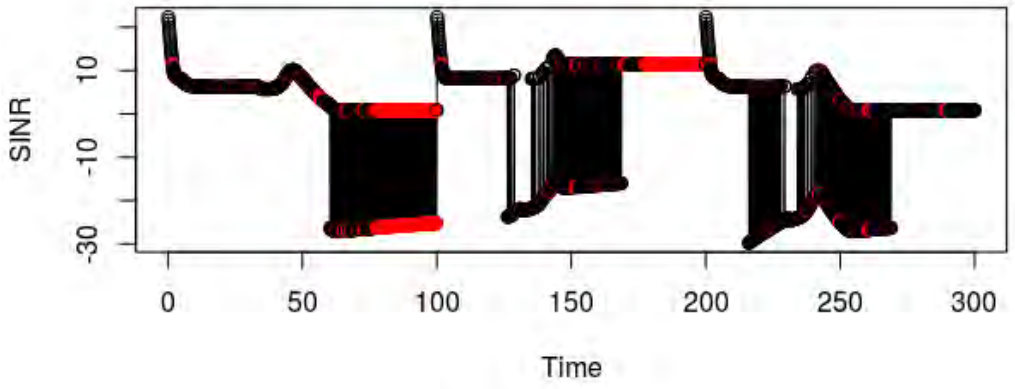

FIGURE 4.2: Clustering results for the initial VRS Algorithm

4.4 Detection System Assumptions

As it is already stated, the machine learning methods that are used are the KNN, the Random Forests and the K-means algorithms respectively. Their

TABLE 4.3: Simulation Parameters

Evaluation Parameters in Veins Simulator	**Values**
$u_{Tx,Rx,Jx}$	15m/sec.
$d_{Tx,Rx}$	35m
$d_{initial}$	200m
P_{T_x,J_x}	100mW
Minimum sensitivity (P_{th})	-86dBm
Transmission Range	130-300 meters

choice does not affect the efficiency of our algorithm as our proposed feature is not constrained by the type of the supervised or usupervised learning algorithm that is used.

Both supervised learning techniques are very popular, with the KNN being robust against noisy training data like the ones obtained from a real-life urban environment and the Random Forests being one of the most accurate algorithms due to the fact that reduces the possibility of overfitting significantly (by averaging several trees, there is a significantly lower chance of overfitting). As far as the unsupervised learning is concerned, the K-means algorithm is one of the fastest and simplest clustering algorithms, while, at the same time, providing easy to interpret clustering results.

Regarding the details of our simulation setup, the *speed* of all the vehicles involved $(u_{Tx,Rx,Jx})$, the initial *distance between the jammer and the pair of* R_x - T_x $(d_{initial})$, the *distance that seperates the receiver from the transmitter* throughout the course of the simulation $(d_{Tx,Rx})$ as well as the power of all the the transmitted signals $(P_{Tx,Jx})$, are presented in Table 4.3.

The power of all the transmitted signals is measured in milliwatts (mW) and is converted in the dBm scale prior to using it in the algorithm. The signal that is transmitted from both the jammer and the transmitter consists of streams that are 500 bits long. For each one of the three scenarios, a number of 1000 packets is transmitted. Using a time sample of 0.1 sec, we simulate the system for 100 seconds (for each scenario) and obtain 1000 measurements (for each scenario).

For the simulation of the movement of the vehicles and their wireless communication we used the Simulation of Urban Mobility (SUMO) and the OMNET++/Veins [26] simulators respectively. SUMO is utilized as our traffic simulator making use of a part of the Erlangen city map while OMNET++ is used to simulate the wireless communication. The evaluation parameters in the Veins simulator are also presented in Table 4.3.

Chapter 5

Simulation Setup

5.1 Jamming Scenarios

In our work, we created three different scenarios - namely **Interference Scenario**, **Smart Attack Scenario** and **Constant Attack Scenario** - each representing a jamming attack case that could, potentially, affect a VANET in real life.

In the *Interference Scenario*, we assume that a jammer is not present in the network so as to evaluate the efficiency of our method in differentiating jamming from interference. The vehicle travels, when, at some point, passes through an area with significant RF interference by which its communication with the other vehicles or with the RSU is affected.

In the *Smart Attack Scenario*, the performance of a more intelligent jammer [13] is evaluated. Specifically a smart jammer starts following the victim-vehicle while transmitting a jamming signal. When the jammer reaches its target at distance of about $10m$, retreats to a safe position and transmits in a reactive way. The jammer is designed to start transmitting upon sensing energy above a certain threshold. The threshold is set to be equal to $-86dBm$, as it is empirically determined to be a good tradeoff between jammer sensitivity and false transmission detection rate. If the detected energy exceeds the threshold during a certain time span of $T_{detection} = 12\mu s$ (during which the jammer observes the energy levels of the channel), an ongoing 802.11p transmission is assumed by the jammer, thus starting its jamming signal transmission for a duration of $T_{duration} = 84\mu s$.

In the *Constant Attack Scenario*, we study the case of a constant jammer that follows the victim-vehicle while transmitting at a minimum power (we chose its initial transmission power to be equal to the $\frac{1}{3}$ of its total power). When the jammer reaches its target, begins transmitting constantly with its full power without any intention to stay undetected.

Fig. 5.1 - 5.3, illustrate the behavior of the jammer by presenting the the plots of SINR versus Time for every one of the three scenarios.

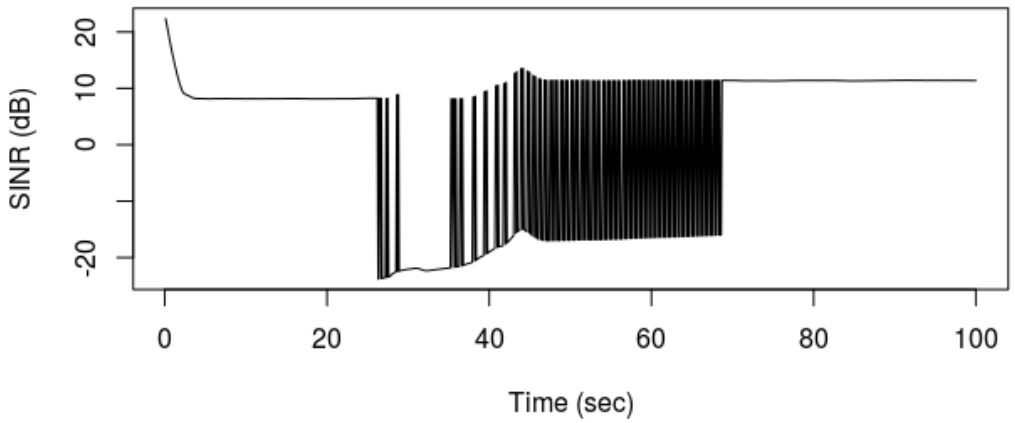

FIGURE 5.1: SINR vs Time for the Rician Fading Model in the Interference Scenario

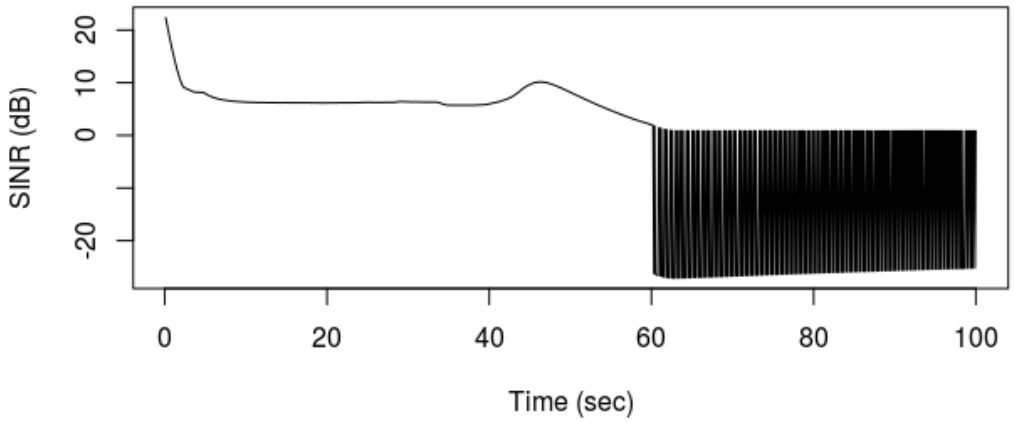

FIGURE 5.2: SINR vs Time for the Rician Fading Model in the Smart Attack Scenario

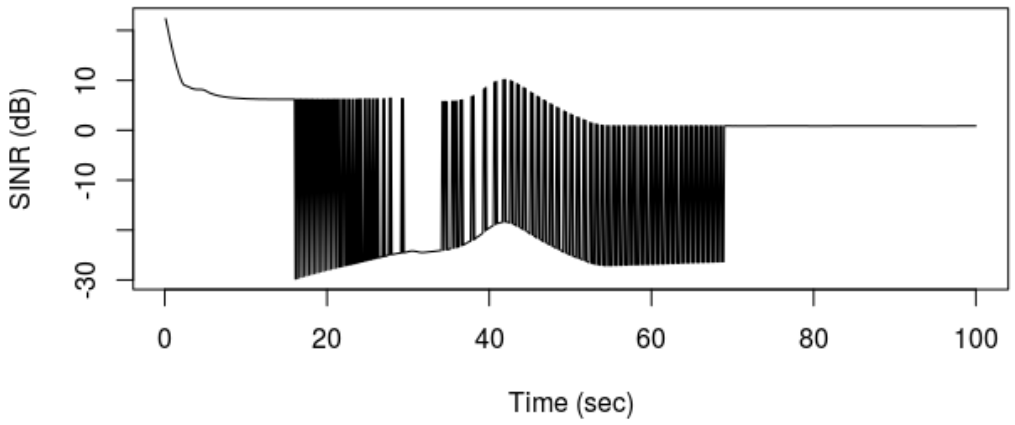

FIGURE 5.3: SINR vs Time for the Rician Fading Model in the Constant Attack Scenario

5.2 Supervised Learning Testing Cases

The previously described scenarios represent possible real-life RF jamming attacks that could affect the V2V or the V2R communication in an urban environment and they are used in both the clustering and the classification process. Beyond that and due to the fact that the classification process is also affected by the training and testing datasets, a series of "Supervised Learning Testing Cases" has been created. These cases allow us to explore deeper the proposed system depending on the on the set of observations that is utilized for training and testing.

These cases only affect how the training and testing is performed, without any further implications in how the scenarios function. They are created in such a way so as to provide us with a comparison between the use or not of the VRS metric in the classification process under various circumstances in all the three scenarios previously presented, that would, in turn, highlight its significance.

- *Train and test with data from the same speed value with the VRS metric **(Same_KNN-VRS and Same_RF-VRS case)**:* the prediction model is trained and tested using observations collected under the speed of 15 m/s, with the use of the VRS metric. To avoid testing with "previously seen data", thus leading to biased classification results, a series of safety measures have been taken to ensure that the training and testing sets are completely separated .

- *Train and test with data from the same speed value without the VRS metric **(Same_KNN and Same_RF case)**:* similar to the previous case, with the only difference being the omission of the VRS metric in the classification process.

- *Train and test under different speed values with the VRS metric* ***(Different_KNN-VRS and Different_RF-VRS case)****:* the previously trained prediction model is tested using data that was collected under a speed of 25 m/s, that is under a speed different from the one the training of the prediction model was based on.

- *Train and test with data from different speed values without the VRS metric* ***(Different_KNN and Different_RF case)****:* similar to the previous case but without the utilization of the VRS metric as an extra feature in the classification process.

- *Train and test with normalized data from the same speed value with the VRS metric* ***(Norm_KNN-VRS and Norm_RF-VRS case)****:* the data is normalized prior to its use training and in testing. By normalization, we refer to the process of changing the data so as to belong in the 0 - 1 range. Both training and testing are conducted on data collected under a speed of 15 m/s but without common observations in the two sets (as stated before).

- *Train and test with normalized data from the same speed value without the VRS metric* ***(Norm_KNN and Norm_RF case)****:* similar to the previous case but without the VRS metric.

Chapter 6

Simulation Results

6.1 Simulation Software

Prior to presenting the unsupervised and supervised learning simulation results, we have to refer to the tools used to perform the classification and the clustering. As it is already stated, the OMNET++/VEINS simulator is used in order to simulate a real-life communication environment from which we can obtain our measurements. To setup and test our clustering and classification based RF jamming attack detection systems on the previously obtained data, we chose to use the programming language R [31].

Although, it is usually used for statistical computing, it provides all the tools (pre-defined algorithms, documentation, graphical representation) required to efficiently use a wide range of machine learning algorithms, thus making it ideal for our purposes. The environment we used is R-Studio [28], an open source, integrated development environment (IDE) for R.

6.2 Unsupervised Learning Simulation Results

Our first approach, regarding the detection of a possible jamming attack affecting the V2V communication, is based on the unsupervised learning algorithm of K-means. It, particularly, utilizes a number of $k = 2$ clusters so as to differentiate an intentional (attack) from an unintentional (interference) jamming scenario, which might be very important for determining the behavior that will be adopted, especially in an urban environment. To prove the importance of the VRS metric in the RF jamming attack detection process, we execute our simulations with and without the use of the new metric and we compare the obtained results.

As it is already stated, each simulation lasts 300 seconds and is equally split in the three attack scenarios discussed in Section 5.1, so that the first 100 sec represent the *Smart Attack scenario*, the next 100 sec the *Interference scenario* and the last 100 sec the *Constant Attack scenario*. The color of the clusters is randomly picked by the visualization tool. A total number of 3000 observations (1000 observations from each scenario), obtained under the speed of 15m/sec ($\approx 54km/h$) and 25m/s ($\approx 90km/h$) respectively, is utilized.

First we begin by examining the observations from the 15 m/sec speed range while using and omitting the proposed VRS metric in the clustering process.

Scenario	Interference Scenario	Smart Attack Scenario	Constant Attack Scenario
Attack	1	1000	1000
Interference	999	0	0

TABLE 6.1: Clustering results for 15m/s using the VRS metric

Scenario	Interference Scenario	Smart Attack Scenario	Constant Attack Scenario
Attack	726	808	672
Interference	274	192	328

TABLE 6.2: Clustering results for 15m/s omitting the VRS metric

The difference between the use and no use of the VRS metric in the clustering process is evident. From Table 6.1 we can see that only one observation is misclassified as intentional jamming while it is actually an observation of the *Interference scenario*. Both for the *Smart Attack scenario* and the *Constant Attack scenario* there are no misclassifications. On the other hand, when we try to differentiate intentional from unintentional jamming while choosing not to use the VRS metric, the "clustering accuracy" [1] drops greatly.

Observing Table 6.2, it is obvious that there is great confusion in distinguishing a jamming attack from interference. From the 1000 observations that actually belong in the *Interference scenario*, only 274 are categorized correctly. The same confusion is, also, observed in the two attack scenarios where there are 192 and 328 wrong categorizations. This could be an important safety issue in an urban environment - **especially when a jamming attack is misinterpreted** [2].

The following figures visuallize the previously described clustering results for the case of a travelling speed equal to 15 m/sec (Fig. 6.1, 6.2). The color of the figures indicates the group in which each observation is clustered to. *The Smart Attack Scenario* and the *Constant Attack Scenario* are represented by the *black* while the *Interference Scenario* by the *red* color. The appearance of more than one colors in each scenario, that is in each 100 seconds (as described in 4.4), indicates the existance of misclassifications:

[1] In fact there is no such term to describe a clustering outcome, but we can use it in our case as it is a priori known that the most accurate clustering result would be close to the one presented previously in Table 6.1.

[2] In that case a behavior not suitable for defending against a malicious RF jamming attack could be wrongly adopted, thus leading to compromisation of safety.

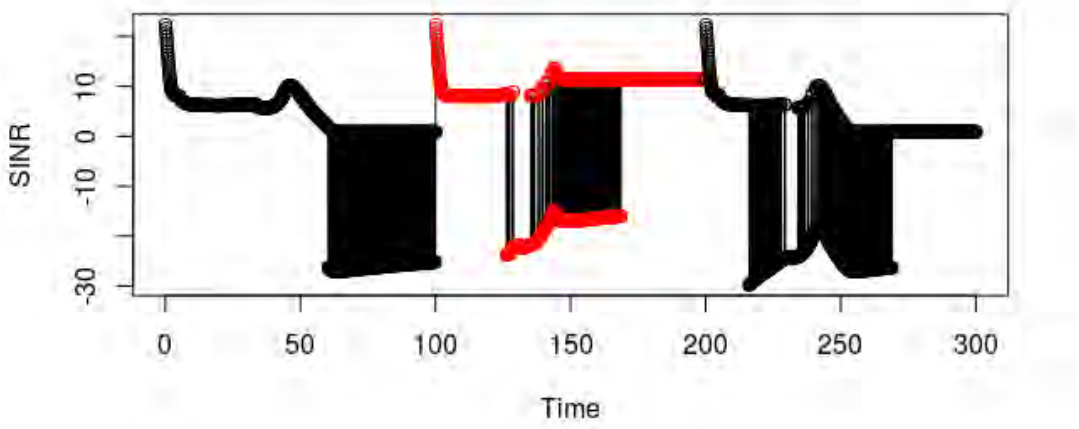

FIGURE 6.1: Clustering results for speed = 15 m/sec with the use of the VRS metric

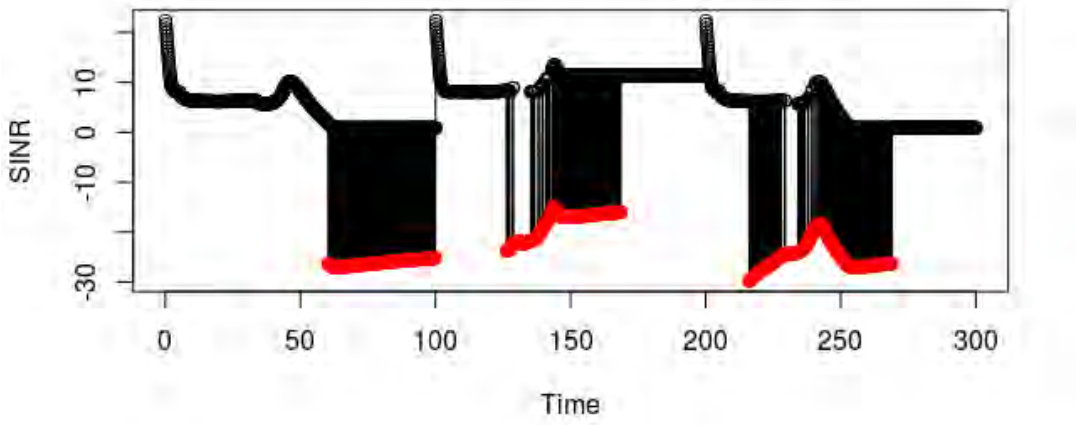

FIGURE 6.2: Clustering results for speed = 15 m/sec without the use of the VRS metric

Having examined the case of medium speed values, we now proceed to a higher speed range that would be much rarer in an urban environment. Nonetheless, it is important to test the performance of the overall system even in this situation. The succeeding tables contain the clustering results from the use and omission of the VRS metric in the clustering process while using observations obtained under a speed of 25 m/sec.

Scenario	Interference Scenario	Smart Attack Scenario	Constant Attack Scenario
Attack	0	1000	1000
Interference	1000	0	0

TABLE 6.3: Clustering results for 25m/s using the VRS metric

Scenario	Interference Scenario	Smart Attack Scenario	Constant Attack Scenario
Attack	719	885	698
Interference	281	115	302

TABLE 6.4: Clustering results for 25m/s omitting the VRS metric

Examining the precedenting tables of results, the importance of the VRS metric in the clustering process is highlighted once more. With the utilization of our proposed metric as an extra feature in the K-means algorithm, we have a perfect differentiation among cases of intentional and unintentional jamming. However, the omission of the feature leads to clustering results similar to the ones presented in Table 6.2, where there is great confusion between the different cases.

This can be visualized in the following figures for the case of a travelling speed equal to 25 m/sec (Fig. 6.3, 6.4):

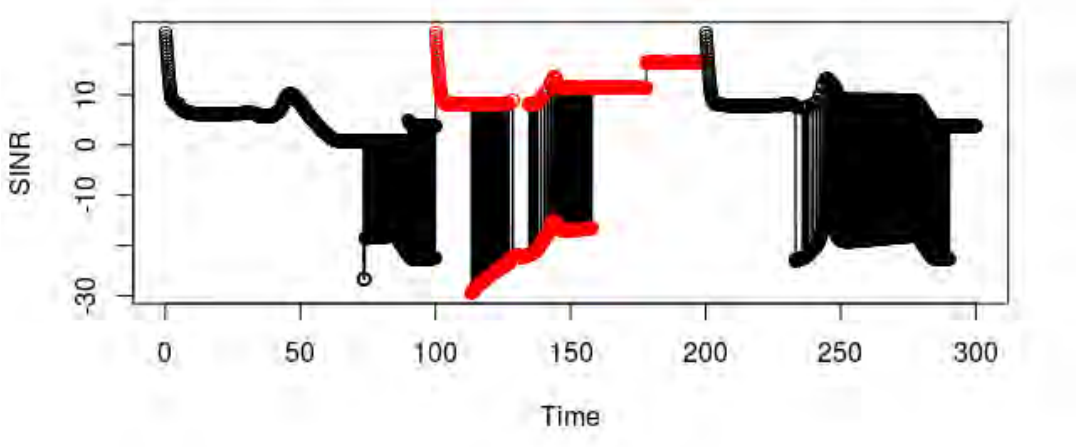

FIGURE 6.3: Clustering results for speed = 25 m/sec with the use of the VRS metric

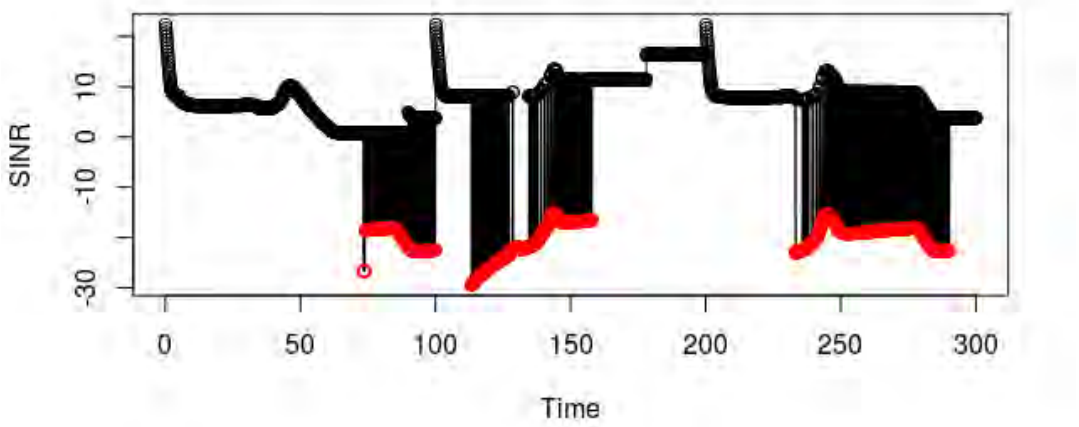

FIGURE 6.4: Clustering results for speed = 25 m/sec without the use of the VRS metric

6.3 Supervised Learning Simulation Results

6.3.1 Simulation Structure: Training and Testing Datasets

We now proceed to our second approach that leverages the use of supervised learning techniques, particularly the KNN and the Random Forests algorithms respectively. To highlight the significance of our proposed system in the classification process, we proceed to comparing the cases presented previously in section 5.2.

In particular, for each case we execute, once more, a simulation which lasts 300 seconds and is equally split in the three jamming scenarios discussed in Section 5.1, so that the first 100 sec represent the Smart Attack Scenario, the next 100 sec the Interference Scenario and the last 100 sec the Constant Attack Scenario. Prior to presenting the classification results, we have to define the size of the training and testing sets as well as the total number of observations used, so as to make them more interpretable.

Each simulation, that is each case from 5.2, uses a set of 3000 observations, equally split into the three attack scenarios examined. To avoid overfitting [3], only 30% of the total number of the observations is used for training while the remaining 70% for testing.

Based on the ratio above, in our simulations, the number of the observations in the training set is 941 whereas the number of the observations in the testing set is 2059 (randomly chosen but almost equally split among the three scenarios in both cases). The following table (Table 6.5) summarizes the number of observations from each scenario, used for training and for testing.

[3]Overfitting occurs when the classifier tends to memorize the training set and thus generalize poorly when facing previously unseen data.

Operation	Interference Scenario	Smart Attack Scenario	Constant Attack Scenario	Total Observations
Testing	703	685	671	2059
Training	293	319	329	941

TABLE 6.5: Number of observations in training and testing datasets

The training process is performed once [4] while the testing takes place either each time a new measurement is obtained or, to avoid high computation cost, at a fixed time instance for a group of previously unseen measurements. This allows our detection system to operate almost in real-time, thus ensuring the identification of an attack and the on-time reaction.

6.3.2 Classification Model Evaluation Measures

The classification results will be presented leveraging the use of a table layout, known as the *confusion matrix*, whose purpose is to visualize the performance of a classification algorithm. Each row of the matrix represents the instances belonging to a predicted class while each column represents the instances in an actual class.

If we have *two classes* in which we want to classify a number of observations, then the confusion matrix is a 2*x*2 table, similar to the precedenting:

		Actual Values	
		Class 1	**Class 2**
Predicted Values	**Class 1**	True Positive	False Negative
	Class 2	False Positive	True Negative

TABLE 6.6: Confusion matrix for two classes

Supposing that the *positive class is Class 1* and the *negative class is Class 2* we conclude that the **True Positive** values are the observations that were *predicted as positive and were actually positive*. Similarly, the **True Negative** values are those *predicted as negative and are actually negative*. On the other hand, the **False Positive** and **False Negative** values are those *predicted as negative/positive and when they are actually positive/negative*. Having more than three classes like in our situation, increases the dimensions of the confusion

[4] In a real-life environment it could be conducted at an initialization phase prior to travelling and if new data were available, otherwise it could be skipped as the training of the predictor would have already been up-to-date.

matrix proportionally. Therefore, in our case, we make use of a $3x3$ confusion matrix with each one of the three classes representing a scenario.

Apart from the mere visualization of the results obtained, the confusion matrix also creates a series of measures that are used in order to determine the performance of the classification model and clarify the obtained results, thus a description of these measures, at this stage, would be important:

- **Accuracy** is the overall accuracy of the classifier, that is all the correctly predicted labels over all the predictions.
- **Sensitivity** or *True Positive Rate* is the proportion of measurements that were predicted to be positive and are positive, from all the measurements that actually are positive (where positive can mean either jamming attack or interference).
- **Specificity** or *True Negative Rate* is the proportion of measurements that were predicted to be negative and are negative, from all the measurements that actually are negative (where negative can mean either jamming attack or interference).
- **Fall-out** or *False Positive Rate* is equal to $1 - TNR$.
- **Miss Rate** or *False Negative Rate* is equal to $1 - TPR$.

Based on the previous definitions, someone could intuitively decide that the measure of accuracy alone would suffice in order to determine the performance of the classification model. In most cases this is true, as it can also be seen from related works like [19] and [6]. However, if the number of observations in different classes varies greatly, then accuracy will not be a reliable metric for the real performance of a classifier, as it will be affected by the class with the larger number of observations and thus a careful inspection of the results is required.

In our case, the number of elements from each scenario that is used in the training and the testing process, is carefully chosen so as to avoid the large data variations described previously. As a result, accuracy alone can be a reliable measure of the classifier performance. However, we will try to explain and justify the calculated accuracy value using the other measures (especially by using the Sensitivity) as well, so as to describe our findings in the clearest way possible.

6.3.3 Rician Fading Model Classification Results

The classification results obtained while simulating the supervised learning testing cases, described in Section 5.2, are presented, in a comparative manner. As it is already stated, these cases are based on the three scenarios from section 5.1 and are created so as to form a comparison between the results obtained from the use and the omission of the VRS metric, respectively, in the classification process.

Comparison of the Same_KNN/RF-VRS and Same_KNN/ RF cases

The classification results for both supervised learning algorithms while using the VRS metric in the classification process, are presented in the following confusion matrices:

Scenario	Interference	Smart Attack	Constant Attack
Interference	703	0	0
Smart Attack	0	494	174
Constant Attack	0	191	497

TABLE 6.7: Confusion matrix for the **Same_KNN-VRS case**

Scenario	Interference	Smart Attack	Constant Attack
Interference	703	1	1
Smart Attack	0	442	167
Constant Attack	0	242	503

TABLE 6.8: Confusion matrix for the **Same_RF-VRS case**

Examining each classification model individually and starting from the one based on the KNN algorithm while using the VRS metric, the calculated accuracy of the prediction model is equal to 82.27%. Moreover, the True-Positive (sensitivity) rate for the Interference Scenario is 100%, for the Smart Attack Scenario 72.12% and for the Constant Attack Scenario 74.07%.

The fairly high true-positive rate (TPR), considering that the data used are close to real-life measurements, thus being affected by the noise of the channel, the travelling speed of the vehicles and the jamming signal that becomes more intense as the jammer approaches it target, yields that fewer intentional jamming attack cases will be undetected or confused with unintentional jamming cases. Apart from that, it can be observed that using the VRS metric in the classification process, leads to limiting the misclassifications among the two intentional jamming attack scenarios with no confusion between jamming attacks and interference. The TPR of each class also justifies the prediction accuracy of the KNN based classification model, which remains unbiased towards the available classes.

Moving on to the classification model of the Random Forests algorithm, the results do not seem to differ significantly. The accuracy of the current classification model is equal to 80.04% and slightly decreases compared to the previous, KNN, case. The TPR of the classes justifies this decrease as for the Interference Scenario is equal to 100%, for the Smart Attack Scenario equal to 64.53% and for the Constant Attack Scenario equal to 74.96%.

With the omission of the VRS metric from the classification process, the results obtained differ significantly from the previous ones, not just in accuracy but also in terms of RF jamming attack and interference differentiation ability.

Scenario	Interference	Smart Attack	Constant Attack
Interference	682	38	33
Smart Attack	17	470	160
Constant Attack	4	177	478

TABLE 6.9: Confusion matrix for the **Same_KNN case**

Scenario	Interference	Smart Attack	Constant Attack
Interference	652	29	23
Smart Attack	23	434	158
Constant Attack	28	222	490

TABLE 6.10: Confusion matrix for the **Same_RF case**

The accuracy achieved by the KNN-based classification model is equal to 79.16%. This might not seem to be a great decrease compared to the KNN-based classification model where the VRS metric was utilized, but observing the results proves the opposite. To understand that, we have to examine the TPR for each class. For the Interference Scenario it is 97.01%, for the Smart Attack Scenario 68.61% and for the Constant Attack Scenario 71.24%. It can be seen that the ability of the classifier to correctly identify cases of intentional or unintentional jamming (TP observations) is affected by the use or omission of the VRS metric. In addition to that, the omission of the VRS metric is highly correlated with the inability to differentiate a jamming attack from interference.

Similar observations can be made based on the outcome of the RF-based classification model, in which case, the accuracy achieved is equal to 76.54%. Examining each class (that is each scenario, as it is already stated) individually, we can see that the TPR for the Interference Scenario is equal to 92.75%, for the Smart Attack Scenario 63.36% and for the Constant Attack Scenario 73.03% respectively. Once more, the exclusion of the VRS metric from the classification process reduces the propability of a correct jamming attack or interference detection, while causing significant confusion among the observations of the different classes (increase of misclassification rate).

The classification results presented above for both cases, can be visualized in the following figures, where the observations of the different classes are highlighted. As described previously in section 6.3.1, each figure represents a simulation that involves all three scenarios *(starting with the Smart Attack*

Scenario, moving on to the Interference Scenario and concluding with the Constant Attack Scenario) and lasts 300 seconds.

Similar to the unsupervised learning approach, the color of the figures indicates the class in which each observation is predicted to belong to. *The Smart Attack Scenario is represented by the red, the Interference Scenario by the black and the Constant Attack Scenario by the green color.* The appearance of more than one colors in each scenario, that is in each 100 seconds (as described in 4.4), indicates the existance of misclassifications.

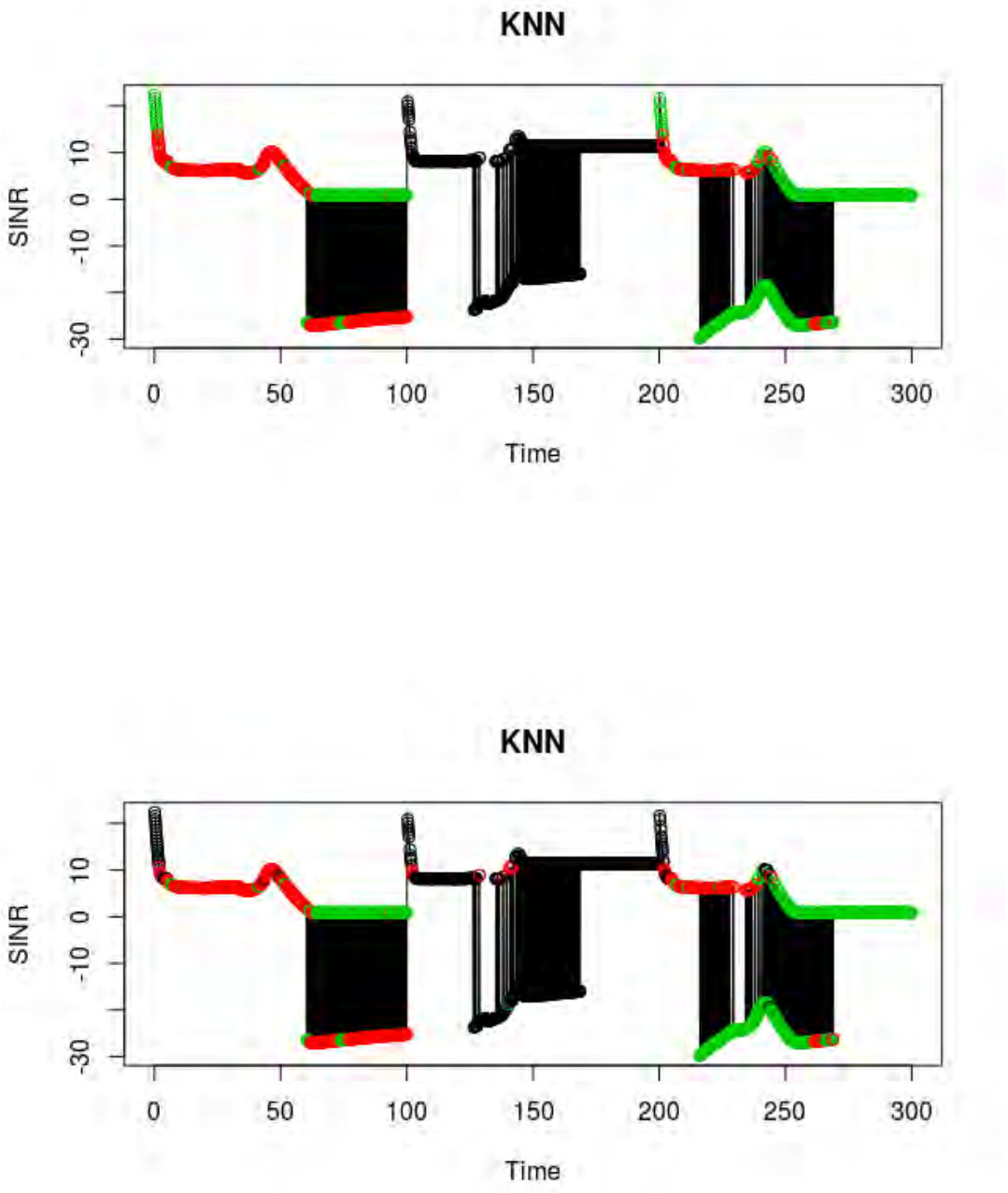

FIGURE 6.5: SINR vs Time for comparing the *Same_KNN-VRS and Same_KNN cases*

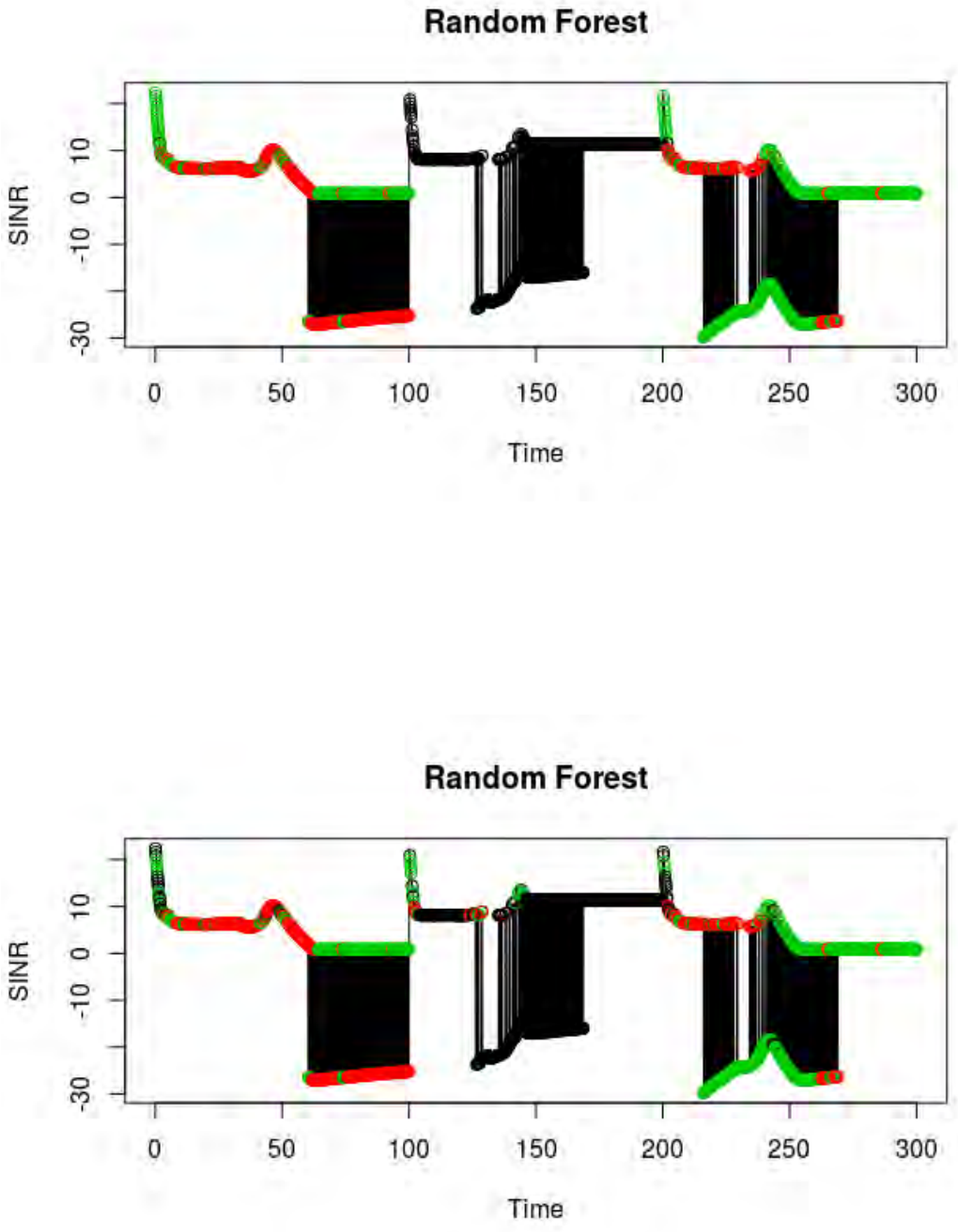

FIGURE 6.6: SINR vs Time for comparing the *Same_RF-VRS and Same_RF case*

Comparison of the Different_KNN/RF-VRS and Different_KNN/RF cases

As stated previously, these cases examine the situation in which the training and the testing are based on observations that were collected under different speed. More specifically, the training of the classifiers (both for the KNN and for the Random Forests algorithm) is conducted based on measurements collected under a speed of $15m/s$. On the other hand, for the testing, measurements collected under a higher speed of $25m/s$ are used. These cases aim at simulating real-life situations in which we might not be able to train and test our classification model on the exact same speed.

We begin the presentation and comparison of the two cases by presenting the classification results obtained while the VRS metric was utilized in both the KNN and the Random Forests algorithms as an extra feature.

Scenario	Interference	Smart Attack	Constant Attack
Interference	663	0	0
Smart Attack	0	458	437
Constant Attack	0	243	258

TABLE 6.11: Confusion matrix for in the **Different_KNN-VRS case**

Scenario	Interference	Smart Attack	Constant Attack
Interference	663	0	1
Smart Attack	0	411	330
Constant Attack	0	290	364

TABLE 6.12: Confusion matrix for the **Different_RF-VRS case**

Examining each classification model individually, we observe that for the KNN-based model the accuracy achieved is equal to 66.97%. In addition, the TPR for the Interference Scenario is equal to 100%, for the Smart Attack Scenario 65.34% and for the Constant Attack Scenario 37.12%. Compared to the repsective results from Table 6.9, where the training and testing measurements were collected under the same speed, a significant decrease can be observed.

This decrease, however, is expected as we choose to test the classification model using data with little similarities compared to the data used for training. Nevertheless, with the use of the VRS metric a clear differentiation between cases of intentional and cases of unintentional jamming is achieved, as well as restriction of the misclassifications between the two, attack scenarios only.

Moving on to the Random Forests-based classification model, we obtain similar results. The accuracy of the classifier is equal to 69.84%, which can

be explained due to the better detection ability in terms of the Constant Scenario. To elaborate, the calculated TPR for the Constant Attack Scenario is equal to 52.37%, while in the KNN-based classifier the respective percentage was only 37.12%. That, in addition to the similar TPR for the Interference and the Smart Attack Scenario (100% and 58.63% respectively) between the two classification models, leads to the increase of the overall accuracy of the classifier.

With the omission of the VRS metric from the classification process, the results obtained differ significantly from the previous ones, not just in accuracy, which is reduced even more, but also in terms of intentional jamming attack and interference differentiation ability.

Scenario	Interference	Smart Attack	Constant Attack
Interference	601	38	380
Smart Attack	11	426	189
Constant Attack	51	237	126

TABLE 6.13: Confusion matrix for the **Different_KNN case**

Scenario	Interference	Smart Attack	Constant Attack
Interference	602	60	212
Smart Attack	13	408	353
Constant Attack	48	233	130

TABLE 6.14: Confusion matrix for the **Different_RF case**

From the confusion matrices presented above, it is evident that the omission of the VRS metric while training and testing using measurements collected under different speed, leads to the worst possible classification outcome. Starting from the KNN-based classifier, the accuracy achieved is equal to 56%, with the TPR for the three classes being 90.65%, 60.77% and 18.13% respectively. The low TPR indicates limited ability in detecting the correct class for each observation. Moreover, it is obvious that there is high confusion between cases of intentional and unintentional jamming, thus making this classifier inadequate for a real-life environment. The same can be stated for the Random Forests-based classification model. The accuracy achieved is equal to 55.37%, with the TPR for the three classes being 90.8%, 58.2% and 18.71% respectively. Once more, not using the VRS metric renders the classification model unable to handle real-life situations.

The results from both cases presented in the respective confusion matrices, can be visualized, with the observations from the different classes being highlighted.

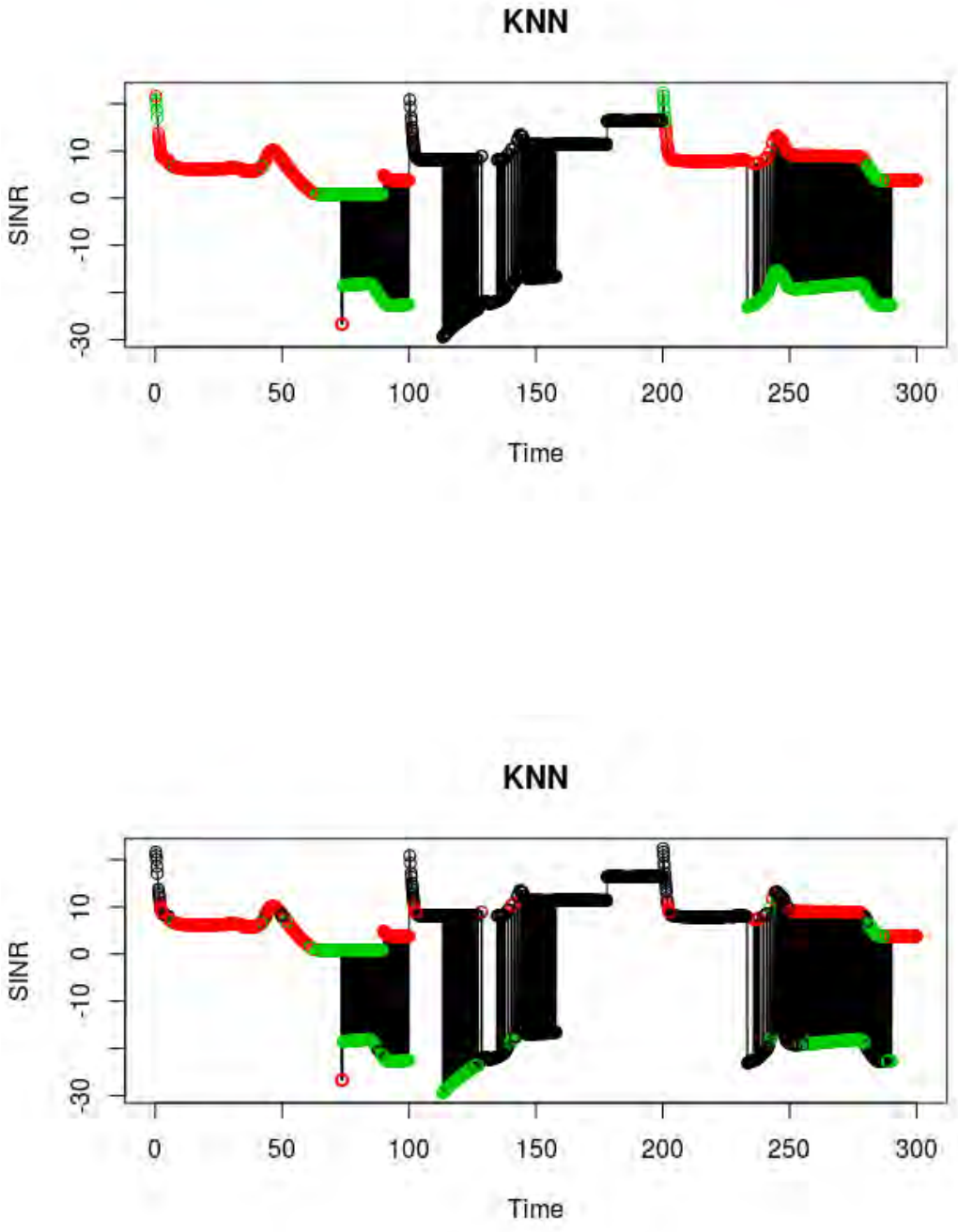

FIGURE 6.7: SINR vs Time for comparing the *Different_KNN-VRS and Different_KNN cases*

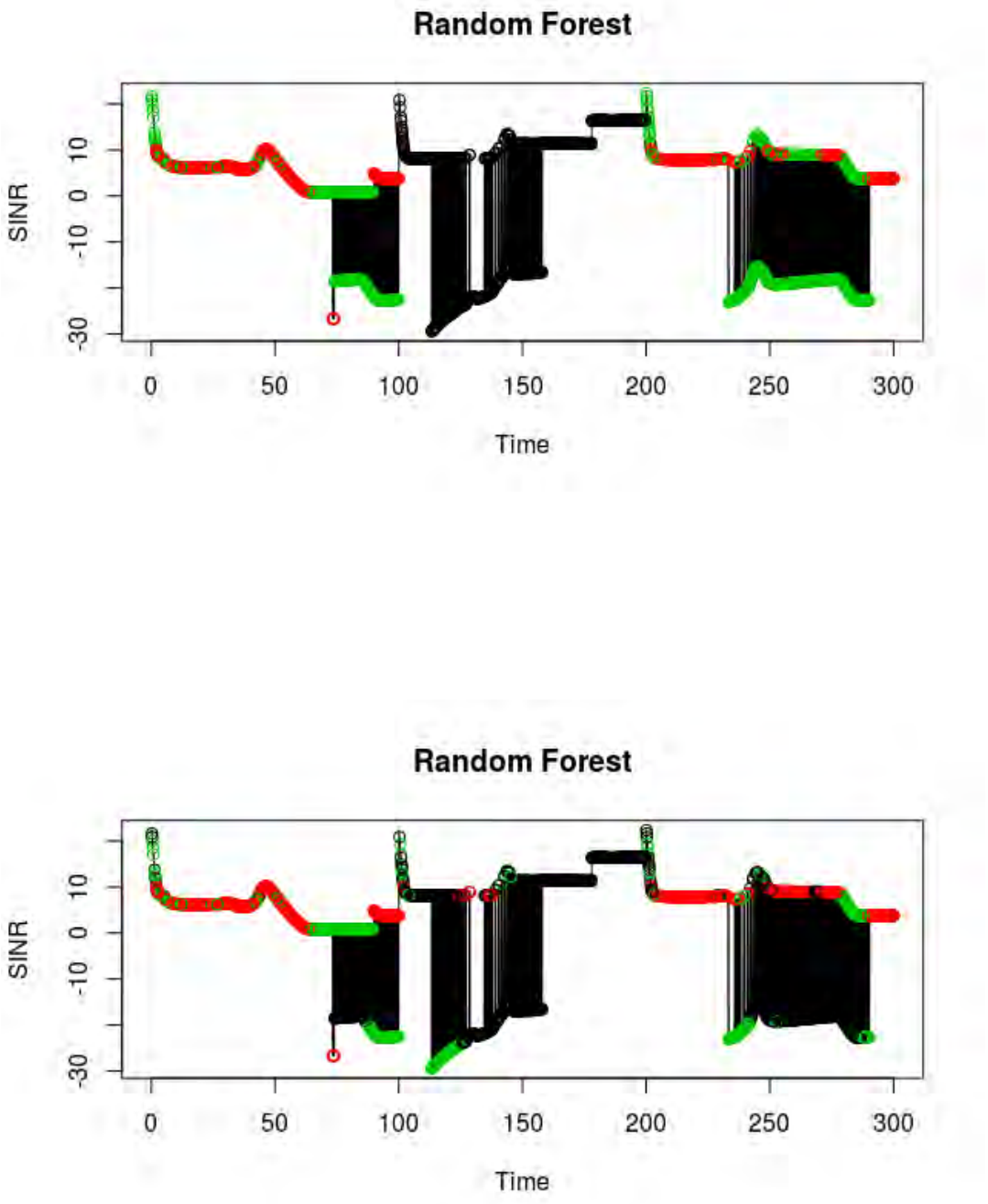

FIGURE 6.8: SINR vs Time for comparing the *Different_RF-VRS and Different_RF case*

Comparison of the Norm_KNN/RF-VRS and Norm_KNN/ RF cases

With these last two cases we aim at determining whether or not the normalization of the data prior to their use in training and in testing affects the classification results, with and without the use of the VRS metric.

Starting from the case in which the VRS metric is utilized as an extra feature in the classification process, the accuracy of the KNN-based prediction model is calculated to be equal to 81.25%. Moreover, examining each class individually we have a TPR equal to 100% for the Interference Scenario, 70.51% for the Smart Attack Scenario and 72.58% for the Constant Attack Scenario. The fairly high TPR for all the classes, indicates an overall good intentional and unintentional jamming differentiation ability, with the misclassifications being limited among the two attack scenarios.

Regarding the Random Forest-based classification model, the accuracy achieved is equal to 80.09%, with the TPR for the three classes being 100% (Interference Scenario), 64.67% (Smart Attack Scenario) and 74.96% (Constant Attack Scenario) respectively. Once more, leveraging the use of the VRS metric enhances the ability of the predictor to detect jamming attack cases and distinguish them from cases of interference.

Apart from that, inspecting the classification results of the respective models (*Same_KNN-VRS* and *Same_RF-VRS* cases) where there is no normalization of the measurements prior to their utilization, we can conclude to that no great difference is observed, neither in terms of accuracy or in terms of sensitivity (TPR). Therefore, we determine that the normalization of the measurements is not required for the efficient operation of the classification models.

The results for both classification models are presented in the following confusion matrices:

Scenario	Interference	Smart Attack	Constant Attack
Interference	703	0	0
Smart Attack	0	483	184
Constant Attack	0	202	487

TABLE 6.15: Confusion matrix for the **Norm_KNN-VRS case**

Scenario	Interference	Smart Attack	Constant Attack
Interference	703	1	1
Smart Attack	0	443	167
Constant Attack	0	241	503

TABLE 6.16: Confusion matrix for the **Norm_RF-VRS case**

As it is evident from all the previous supervised learning testing cases, the exclusion of the VRS metric leads to an overall deterioration of the classification outcome - both in terms of accuracy and in terms of successful and efficient detection and differentiation ability.

Starting from the KNN-based predictor, the accuracy achieved is equal to 78.1%. Compared to the previous case where the VRS metric was utilized, we observe that the drop is not so significant. The same is true for the TPR values of the three classes. The calculated TPR for the Interference Scenario is 95.87%, for the Smart Attack Scenario 67.3% and for the Constant Attack Scenario 70.49%.

Where the difference lays, however, is in the misclassifications. In the current case there are several misclassifications that confuse, once again, cases of intentional jamming with cases of interference, while, on the contrary, with the use of the VRS metric these errors were confined among the two attack scenarios. This can also be seen from the results of the Random Forests-based predictor, whose accuracy is equal to 76.4%. The following confusion matrices present the results for the two classification models:

Scenario	Interference	Smart Attack	Constant Attack
Interference	674	42	34
Smart Attack	29	461	164
Constant Attack	0	182	473

TABLE 6.17: Confusion matrix for the **Norm_KNN case**

Scenario	Interference	Smart Attack	Constant Attack
Interference	651	28	21
Smart Attack	27	436	164
Constant Attack	25	221	486

TABLE 6.18: Confusion matrix for the **Norm_RF case**

The visualization of the results is conducted as previously and presented in the following figures:

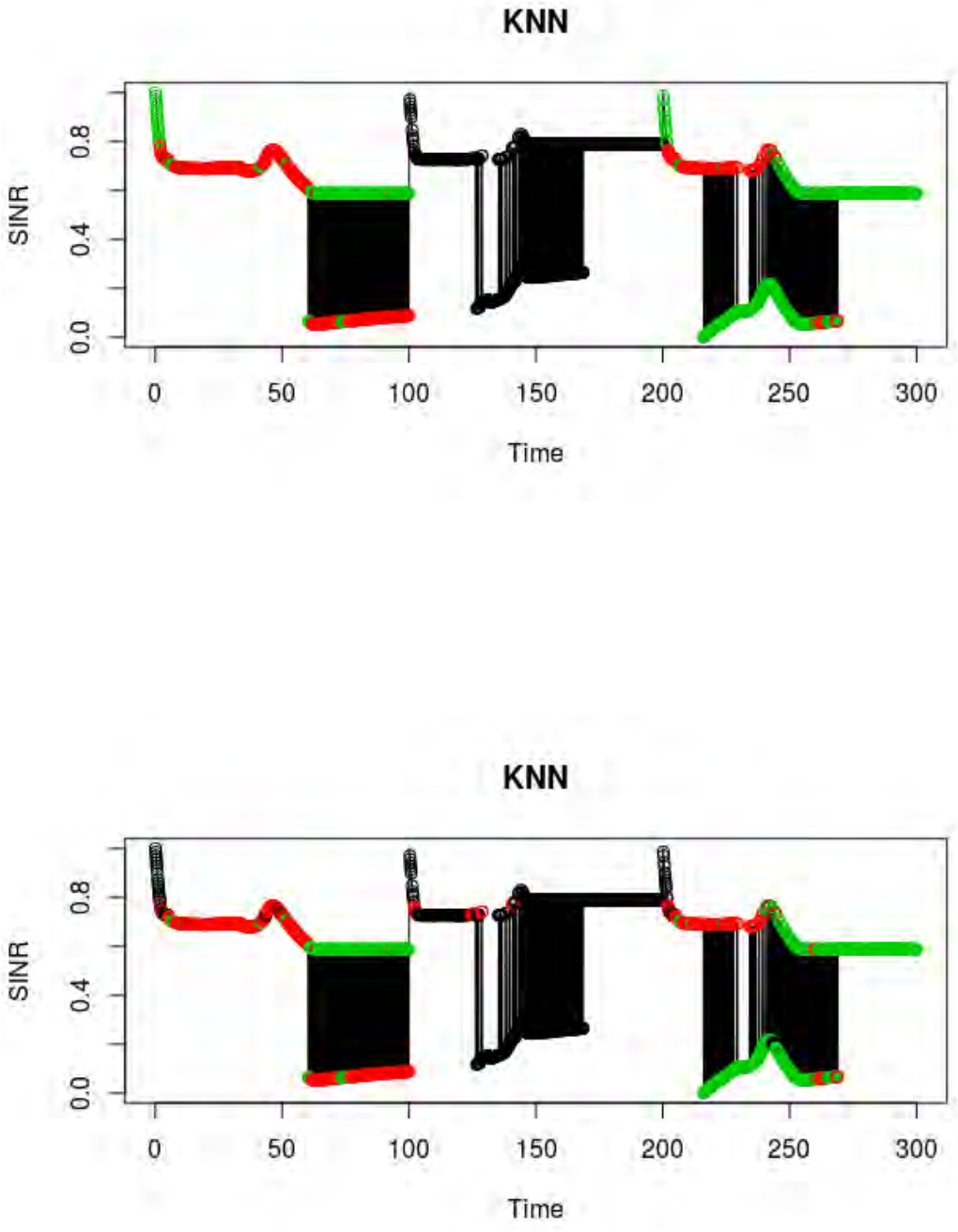

FIGURE 6.9: SINR vs Time for comparing the *Norm_KNN-VRS and Norm_KNN cases*

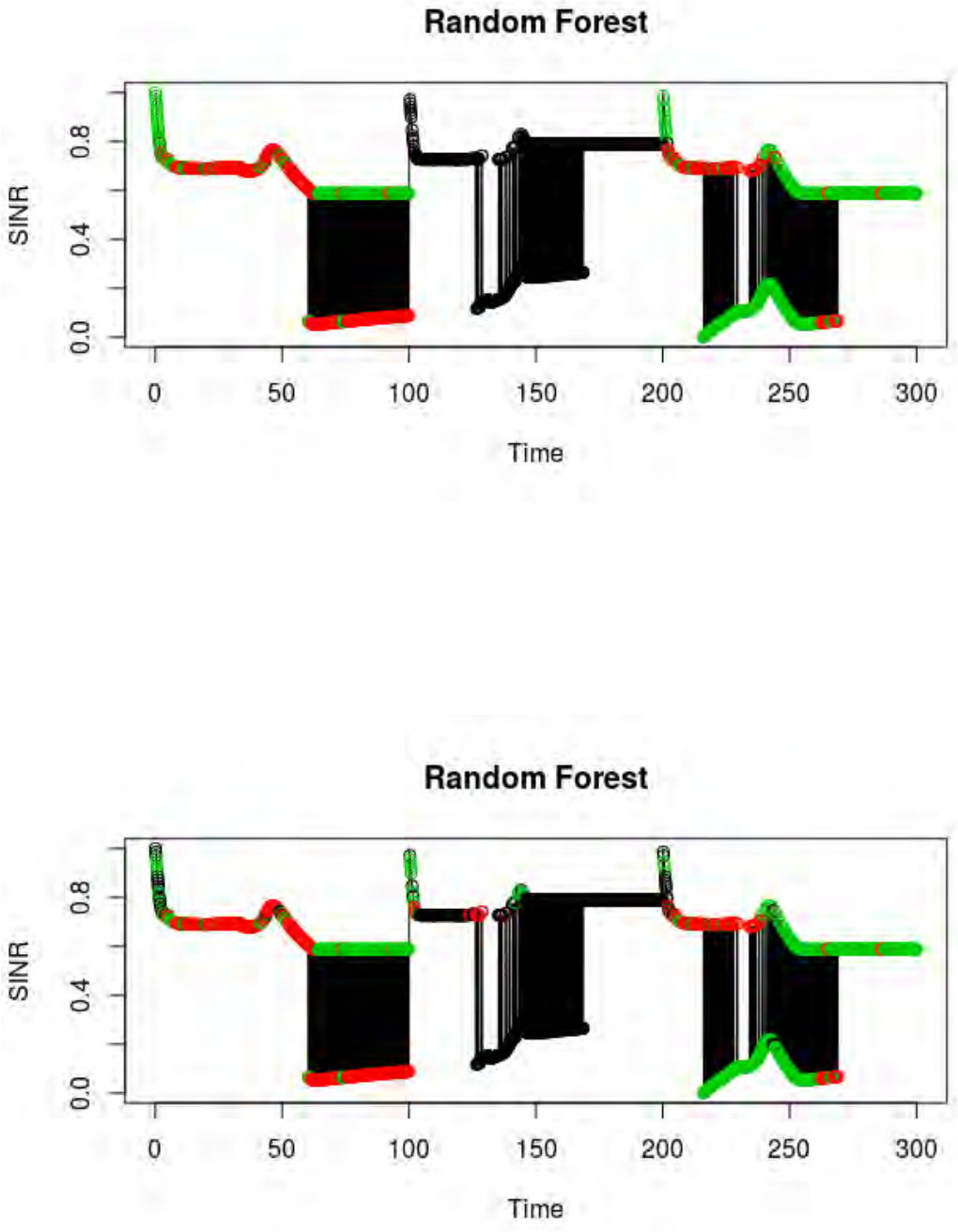

FIGURE 6.10: SINR vs Time for comparing the *Norm_RF-VRS and Norm_RF case*

6.3.4 Classification Accuracy Synopsis

Fig. 6.11 summarizes the accuracy achieved by both the KNN and the Random Forests algorithm when based only on the features previously used in the literature for jamming attack detection [19], compared to the proposed approaches KNN-VRS and RF-VRS that use the VRS metric. The VRS metric increases the accuracy of the classifier and ensures almost perfect differentiation between cases of intentional and unintentional jamming. When using the VRS metric while training and testing with data from the same speed, there is an increase up to about 4% in the classification accuracy. On the other hand, when testing with data from a different speed compared to the one the training was based on, the increase in the classification accuracy is even greater, reaching up to about 14%.

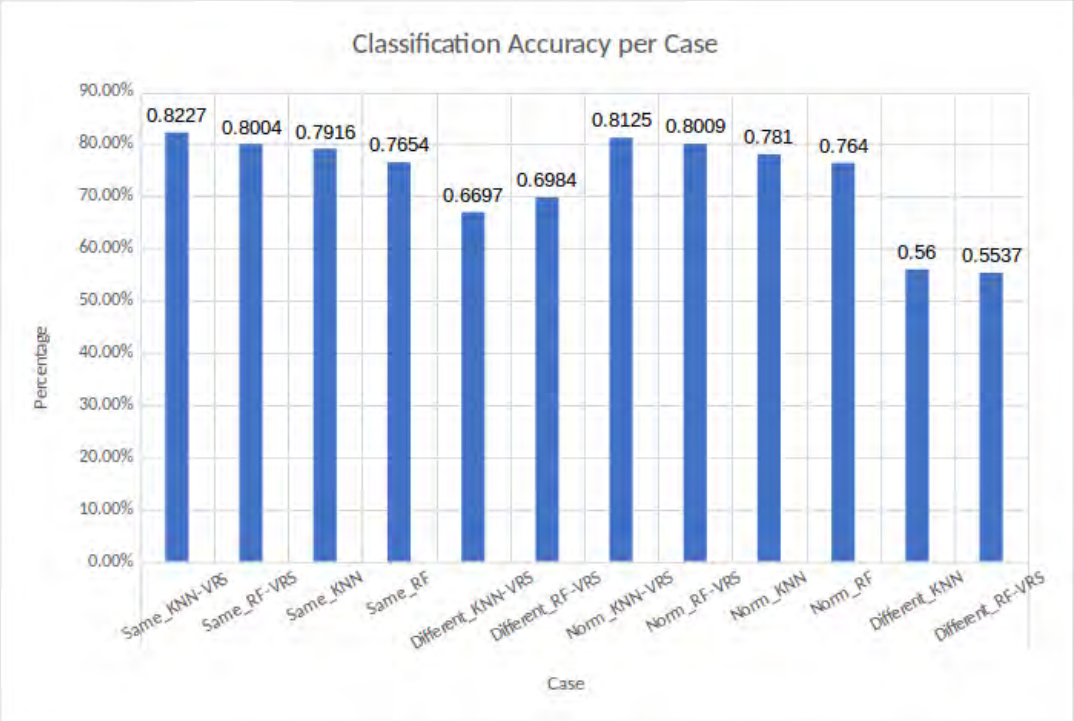

FIGURE 6.11: Achieved accuracy of the classification model when using or omitting the VRS metric

Chapter 7

Conclusion

In this work, we presented a scheme for detecting a specific type of DoS attack, namely RF jamming attack, that is based on cross-layer supervised and unsupervised machine learning and that exploits a novel metric from the application layer, the variations of the relative speed between the jammer and the target vehicles. The relative speed is passively estimated from the combined value of the desired and the jamming signal at the target vehicle. To evaluate the significance of the proposed feature, we implemented three different scenarios - two with a jammer present and one with interference only.

With our work, we introduced a proactive approach against potential RF jamming attacks which is able to differentiate benign from malicious jamming, that is interference from jamming, distinguish the unique characteristics of each attack, if there are more than one types of jammer affecting communication and predict a threat, thus minimizing the security risks for two or more connected vehicles.

We also showed that typical wireless receiver measurements from the network and the physical layer like PDR, SINR, RSS might be able to detect an attack but are less efficient in distinguishing interference from intentional jamming, which is very important in an urban environment. Through our evaluation, using the proposed classification algorithms - namely KNN-VRS and RF-VRS respectively - we were able to point out the vital role of the relative speed and its variations from the application layer in jamming detection and unintentional jamming cases differentiation, as well as in the overall increase of the accuracy achieved by the classification model.

Bibliography

[1] Mani Amoozadeh et al. "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving". In: *IEEE Communications Magazine* 53.6 (2015), pp. 126–132.

[2] Meysam Azizian, Soumaya Cherkaoui, and Abdelhakim Senhaji Hafid. "Link Activation with Parallel Interference Cancellation in Multi-Hop VANET". In: *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th* (2016).

[3] Ikechukwu K Azogu et al. "A new anti-jamming strategy for VANET metrics-directed security defense". In: *Globecom Workshops (GC Wkshps), 2013 IEEE*. IEEE. 2013, pp. 1344–1349.

[4] Norbert Bißmeyer, Christian Stresing, and Kpatcha M Bayarou. "Intrusion detection in vanets through verification of vehicle movement data". In: *Vehicular Networking Conference (VNC), 2010 IEEE*. IEEE. 2010, pp. 166–173.

[5] Darren Cottingham. *What is vehicle platooning? Driving Tests*. https://www.drivingtests.co.nz/resources/what-is-vehicle-platooning/. 2017.

[6] Jyoti Grover et al. "Machine Learning Approach for Multiple Misbehavior Detection in VANET". In: *International Conference on Advances in Computing and Communications* 192 (2011), pp. 644–653.

[7] Ali Hamieh, Jalel Ben-Othman, and Lynda Mokdad. "Detection of radio interference attacks in VANET". In: *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE. 2009, pp. 1–5.

[8] John A Hartigan and Manchek A Wong. "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1 (1979), pp. 100–108.

[9] John A Hartigan and Manchek A Wong. "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds". In: *Internet of Things (WF-IoT), 2014 IEEE World Forum* (2014).

[10] Hamssa Hasrouny et al. "VANet security challenges and solutions: A survey". In: *Vehicular Communications* (2017).

[11] Alex Herm. *Assume self-driving cars are a hacker's dream? Think again*. https://www.theguardian.com/technology/2017/aug/30/self-driving-cars-hackers-security. [Online; accessed 30-August-2017]. 2017.

[12] Joshua Jo and Mario Gerla. "Internet of Vehicles and Autonomous Connected Car - Privacy and Security Issues". In: *Computer Communication and Networks (ICCCN), 2017 26th International Conference* (2017).

[13] Dimitrios Kosmanos et al. "MIMO Techniques for Jamming Threat Suppression in Vehicular Networks". In: *Mobile Information Systems* 2016 (2016).

[14] Andy Liaw, Matthew Wiener, et al. "Classification and regression by randomForest". In: *R news* 2.3 (2002), pp. 18–22.

[15] Sharaf Malebary, Wenyuan Xu, and Chin-Tser Huang. "Jamming mobility in 802.11 p networks: Modeling, evaluation, and detection". In: *Performance Computing and Communications Conference (IPCCC), 2016 IEEE 35th International*. IEEE. 2016, pp. 1–7.

[16] Lynda Mokdad, Jalel Ben-Othman, and Anh Tuan Nguyen. "DJAVAN: Detecting jamming attacks in Vehicle Ad hoc Networks". In: *Performance Evaluation* 87 (2015), pp. 47–59.

[17] Anh Tuan Nguyen, Lynda Mokdad, and Jalel Ben Othman. "Solution of detecting jamming attacks in vehicle ad hoc networks". In: *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*. ACM. 2013, pp. 405–410.

[18] Oscar Puñal, Ana Aguiar, and James Gross. "In VANETs we trust?: characterizing RF jamming in vehicular networks". In: *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*. ACM. 2012, pp. 83–92.

[19] Oscar Puñal et al. "Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation". In: *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*. IEEE. 2014, pp. 1–10.

[20] Abdul Quyoom et al. "A novel mechanism of detection of denial of service attack (DoS) in VANET using Malicious and Irrelevant Packet Detection Algorithm (MIPDA)". In: *Computing, Communication & Automation (ICCCA), 2015 International Conference on*. IEEE. 2015, pp. 414–419.

[21] S RoselinMary, M Maheshwari, and M Thamaraiselvan. "Early detection of DOS attacks in VANET using Attacked Packet Detection Algorithm (APDA)". In: *Information Communication and Embedded Systems (ICICES), 2013 International Conference on*. IEEE. 2013, pp. 237–240.

[22] Mohammed O Shafiq et al. "Detection and prevention of distributed denial of services attacks by collaborative effort of software agents, first prototype implementation". In: *Parallel and Distributed Computing and Networks: Proceedings of the 23rd IASTED International Multi Conference on Applied Informatics*. IASTED. 2005.

[23] Mehreen Shaikh and Abid H Syed. "A SURVEY ON JAMMING ATTACKS, DETECTION AND DEFENDING STRATEGIES IN WIRELESS SENSOR NETWORKS". In: ().

[24] Mehreen Shaikh and Abid. H Syed. "Jamming sensor networks: attack and defense strategies". In: *IEEE Network* 20.3 (2006), pp. 41–47.

[25] S Sharanya and S Karthikeyan. "CLASSIFYING MALICIOUS NODES IN VANETS USING SUPPORT VECTOR MACHINES WITH MODIFIED FADING MEMORY". In: (2006).

[26] Christoph Sommer, Reinhard German, and Falko Dressler. "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis". In: *IEEE Transactions on Mobile Computing* 10 (1) (2015), pp. 3–15.

[27] Oliver Sutton. "Introduction to k nearest neighbour classification and condensed nearest neighbour data reduction". In: *University lectures, University of Leicester* (2012).

[28] *Take control of your R code*. https://www.rstudio.com/products/rstudio/. Accessed: 2017-11-12.

[29] David Tse and Pramod Viswanath. "Fundamentals of Wireless Communication". In: *Cambridge University Press*. 2005.

[30] DAISUKE WAKABAYASHI. *Waymo's Autonomous Cars Cut Out Human Drivers in Road Tests*. https://www.nytimes.com/2017/11/07/technology/waymo-autonomous-cars.html. [Online; accessed 7-November-2017]. 2017.

[31] *What is R?* https://www.r-project.org/about.html. Accessed: 2017-10-30.

[32] Wenyuan Xu et al. "The feasibility of launching and detecting jamming attacks in wireless networks". In: *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. ACM. 2005, pp. 46–57.